

ONTOLOGICAL ELEMENTS FOR INFORMATION SYSTEMS ARCHITECTURE

Harry Katzan, Jr.
Savannah State University
katzanh@savstate.edu

Abstract

An information systems architecture is a specification of the information, computational elements, people, and underlying concepts in a services system. Accordingly, an associated data architecture is a specification of how data is stored, processed, and used in a system of record. A given data architecture effectively governs the flow of data in a system and establishes a relationship between an organization's processes and its supporting information systems. Data architecture for an enterprise describes its applications, hardware, networks, business objects, business processes, and data. As such, it encompasses a complex set of definitions, methods, data types, and relationships. This paper introduces the concept of ontology applied to information systems and describes the essential elements of data architecture.

Keywords: Data architecture, ontology, business objects, framework

Ontology

In philosophy, ontology is the study of being (or existence), describes the basic categories, and defines the entities and classes of elements in its domain. In the area of information systems, it is a specification of conceptualization used to enable knowledge sharing and reuse. More specifically, an ontology can be viewed as a data model that describes objects, classes, attributes, and relations. In his ground-breaking book on knowledge representation, John F. Sowa gives an appropriate definition for our purposes:

The subject of *ontology* is the study of the *categories* of things that exist or may exist in some domain. The product of such a study, called an *ontology*, is a catalog of the types of things that are assumed to exist in a domain of interest D from the perspective of a person who uses the language L for the purpose of talking about D . The types in the ontology represent the *predicates*, *word senses*, or *concept and relation types* of the language L when used to discuss topics in the domain D . (Sowa, 2000)

One common approach to the delineation of ontological elements is to divide the extant entities into groups called "categories." These lists of categories are quite different from one another. It is in this latter sense that ontology is applied to such fields as theology, information science, and artificial intelligence. (Wikipedia 2007)

Ontological Engineering

Ontological engineering encompasses a set of activities conducted during conceptualization, design, implementation, and deployment of ontologies. (Dedvedzic, 2002) Ontological engineering seeks to achieve the following goals in a given domain:

- Definition of terms
- Establishment of a body of domain knowledge
- Specification of coherent and expressive knowledge bases

In short, an ontology defines the vocabulary of a problem domain and a set of constraints on how terms are related. It also gives data types and operations defined over the data types.

Data Architecture

Data architecture describes how data is processed, stored, and utilized in a given system. It provides criteria for data processing operations that make it possible to design data flows and also control the flow of data in the system. (Wikipedia, *op. cit.*) Stephens defines data architecture as a corporation's expression of strategy for creating and managing the use of data in order to transform data into information. He goes on to note that it is a strategic asset that must assure the following:

- Standardization of data structures
- Definition protection of the data resource
- Consistence and quality of the data resource
- Judicial use of corporate resources
- Delivery of timely and credible data
- It addresses business needs
- That it can be managed in a corporate environment

The fact that data architecture is a “grey architecture” does not detract from its validity. (Stephens, 2007)

Business Objects

A *business object* encapsulates the business logic of a single business entity and the data pertinent to that entity. (Lewis, 2001) Each business object consists of one or more service components and one or more data components. A *service component* is the functionality supplied by a business object. A *data component* encapsulates data stored in a database.

Accordingly, data architecture is conceptualized as a set of business objects.

Frameworks

A *framework* is a taxonomy for looking at a class of computer applications. It effectively works by identifying the pieces that constitute an application and prescribing how they should be put together. Typically, a framework emphasizes segmentation and sharing at the data level. Operational and reference data are separated, so that reference data may be shared among frameworks. Rather than viewing an application as a monolithic structure supporting a particular business function, it is useful to view an application framework as a collection of one or more of the following elements: a computational engine, a data management component, application building blocks, and a user-interface module. This viewpoint recognizes and takes advantage of off-the-shelf components. The needed infrastructure includes a reference data architecture and a metadata repository. A *reference data architecture* is a prescription of how data should flow within a single application framework and between diverse application frameworks. A *metadata repository* is the concept that allows middleware to transfer data between platforms on a scheduled or dynamic basis.

A framework can be conceptualized as a collection of three views: organization, business, and technical. The *business view* is significant and consists of data, function, workflow, and the solution structure. Within this view, the *data* refers to the operational and historical databases, the legacy files, the staging database, and the transfer files. The *function* denotes the applications and the programs that transfer data between applications. The *workflow* refers to the behavioral aspects of the combination of data and function. The

solution structure is the data architecture that holds information about the combination of generic components from the data, function, and workflow areas.

Application Domains and Application Groups

An *application domain* (AD) is a suite of application programs, user interfaces, databases, reporting facilities, file, data marts, and related entities that collectively perform an enterprise function. An example would be “Order Management and Trading.” In an application domain, all applications are related but some applications are more related than others. Related ADs communicate with each other through a *virtual data exchange* (VDE). The notion of a VDE stems from the need to avoid the case where ADs communicate directly. A set of ADs connected by a VDE are called an *application group* (AG), as suggested by the example given in Figure 1. An example of an AG would be “Investment Operations,” perhaps consisting of the following ADs: Order Management and Trading, Holdings Management, Investment Accounting, Custody Processing, and Back Office Integration. The primary benefit of this viewpoint is that an application group maintains its autonomy while participating in the overall enterprise design through the reference data architecture.

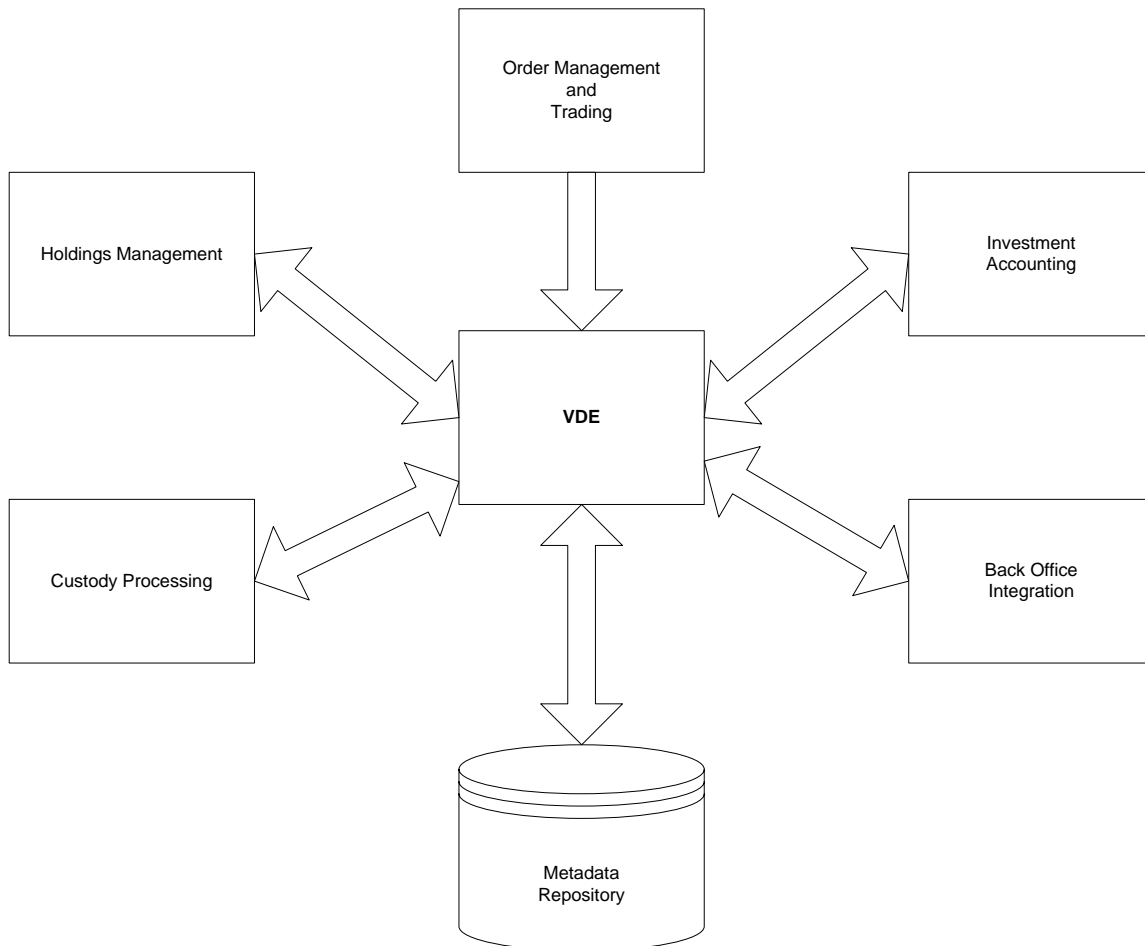


Figure 1. Application Group

Communicating between Application Groups

Information systems applications are grouped because they possess well-defined operational properties in common. For example, it is known beforehand that certain data from a risk management application are

used in a portfolio construction application. Moreover, it is almost always the case that immediate data are needed, so that the exchange of data must be performed on a real-time basis. This is a defining characteristic of data exchange within an application group. By definition, the data exchange between Application Groups is not on a real-time basis – otherwise, the constituent ADs would be in the same AG. The data exchange modality between AGs is the enterprise data warehouse. The *enterprise data warehouse* is comprised of the reference and historical databases that are needed to run the business, but at the same time, are of such a nature that they can be shared between Application Groups. Figure 2 gives a sample investment data architecture.

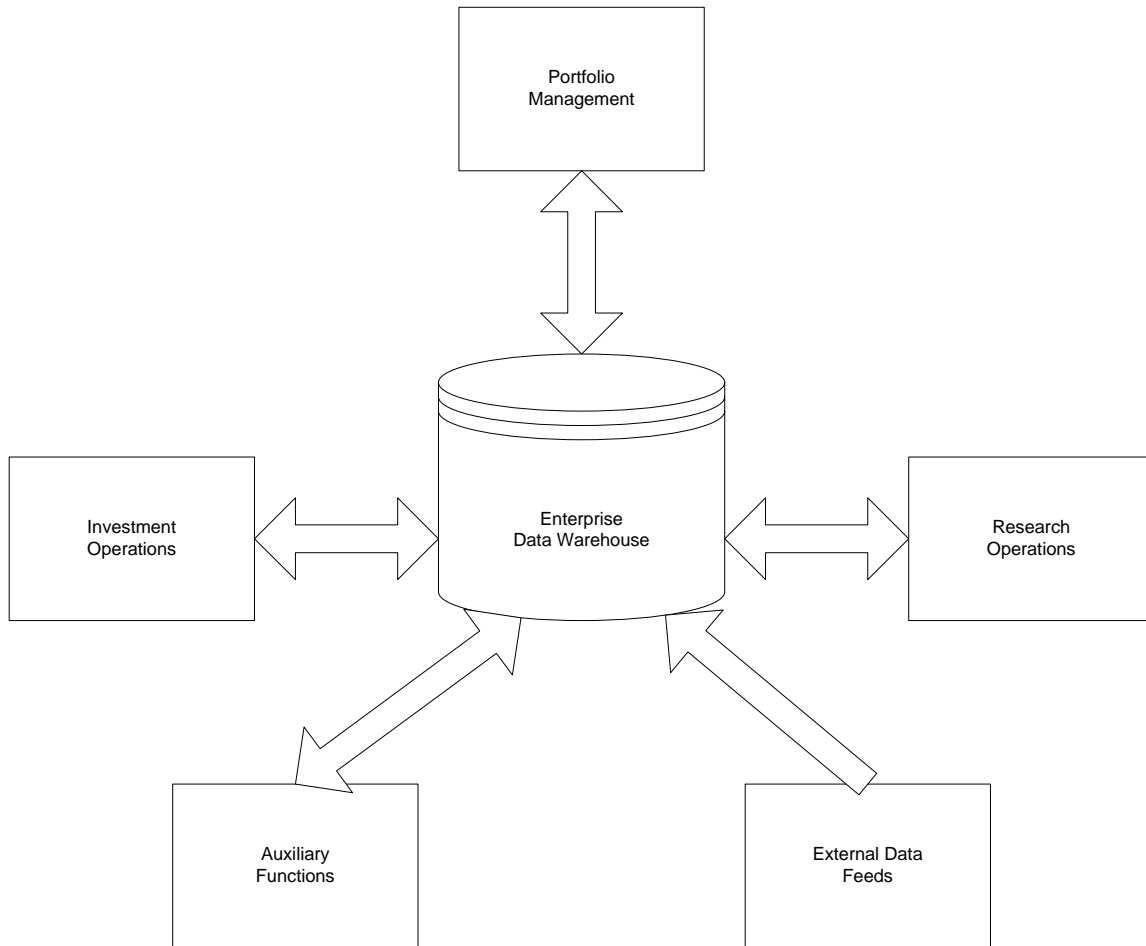


Figure 2. Sample Information Architecture

Data Architecture Ontology

The preceding sections give the context for the ontology of data architecture. This section gives definitions for the ontological elements, as listed in Table 1.

1. **Data view** is a collection of single elements, such as name, address, order number, etc. Data elements are commonly structured for storage and retrieval purposes and are normally associated with an application or a database, requiring an appropriate semantic context.
2. **Information view** is data to which knowledge has been applied. This view incorporates an appropriate semantic definition and represents a “business view” of data. Information is commonly aggregated for analysis, inference, understanding, and decision making.

3. **Data architecture** reflects the structure of storage components and the manner in which data is exchanged between them. Data architecture also defines how data will be managed, organized, stored, transformed, accessed, and otherwise made available to applications and users. Data architecture is composed of conceptual-level and logical-level descriptions and derives its applicability through a framework.
4. **Conceptual-level data architecture** defines design constructs, known as templates, and best-use principles concerning their use. It also describes standards, policies, and processes for use of subject data elements.
5. **Logical-level data architecture** applies the conceptual constructs and processes to an application domain, and establishes and incorporates data management structures and technology.
6. A **Framework** is a collection of components, connections, actors, and use cases that comprise a reusable design of a data architecture for a class of business problems. A framework works with a subject data model to establish a context for a data architecture.
7. **Information architecture** represents the business-process view of a data architecture, and delineates the information needed to support different business processes. A framework is used to relate technology to users, information structures, and use cases.
8. **Enterprise data architecture** is a conceptual-level data architecture that is reusable across the enterprise. This element facilitates data sharing business units and enables risk avoidance and cost reduction.
9. **Subject data model** is a high-level view of data with a business perspective, and represents an aggregate view of information from multiple application systems. This data model supports a CRUD matrix integration of application and infrastructure.
10. **Operational database** is an online system that usually supports transaction processing and supports operational queries and update capability by being "current" data valued and subject oriented.
11. **Data warehouse** is a historical database that is organized for efficient updating from source systems. It may incorporate aggregation and summarization to support upstream data marts. A data warehouse may also allow access/reporting but is not usually optimized for that.
12. **Data mart** is a historical database optimized for access and reporting around a business unit. It is usually updated from a data warehouse via "batch updates."
13. **Metadata** is a common repository of "data about data" that is sharable between applications and platforms, reflecting business semantics and data sharing.
14. **Staging database** is a focal point for data cleansing operations on dynamic data from operational databases.
15. **Data exchange** is an event driven platform for exchanging data between applications.
16. **Data hub** is a middleware system designed to move data between source and target systems. The source and target systems include operational databases, legacy files, and data warehouse/mart data stores.
17. **Reference data architecture** refers to goal-state architecture, enterprise data warehouse, department data mart, and reporting systems.

Collectively, the definitions provide a means of communicating in the problem domain of data architecture.

TABLE 1. Ontological Elements

Number	Element
1	Data View
2	Information View
3	Data Architecture
4	Conceptual-Level Data Architecture
5	Logical-Level Data Architecture
6	Framework
7	Information Architecture
8	Enterprise Data Architecture
9	Subject Data Model
10	Operational Database
11	Data Warehouse
12	Data Mart
13	Metadata
14	Staging Database
15	Data Exchange
16	Data Hub
17	Reference Data Architecture

Delineation of Current Business Problems and Ongoing Development

Current business problems, at least from the boardroom, could be characterized as follows:

- Inconsistent and unreliable results
- Reports cannot be obtained quickly enough
- Needed information for decision making not readily available
- Data analysis is often performed on an ad hoc basis
- New applications are delayed
- Incompatibility systems

From an IT perspective, some of the causes can be identified, almost by inspection:

- Data is moved and copied many times
- Production systems are used as sources for data analysis and reporting
- Weak tools are employed for data analysis
- Data management is not aligned with business functions
- Integration of legacy systems with new applications

Accordingly, it would seem appropriate that ongoing research should concentrate on both business and IT solutions. From a business perspective, here is a sample set of solution elements:

- Implement data movement, data modification, and messaging through managed facilities
- Separate decision support, data analysis, and reporting functions from operational systems where data is constantly changing
- Align application development with business units
- Standardize business units on data access and reporting tools and data analysis facilities

Naturally, IT will respond with “we already do it” and you are not aligned with it, or “we can do it,” if we had the resources. Perhaps, here is what we have in mind through a formalized data architecture:

- Establish system of record procedures
- Perform data extraction, transformation, and load through data hubs
- Enforce metadata repository rules to assist in inter-application data exchange
- Use data warehouse/mart facilities for access, reporting, and analysis
- Adopt technology standards for data/application design

To summarize, perhaps we should focus on *Systems of Data* before we focus on *Systems of Services*.

References

http://en.wikipedia.org/wiki/Data_Architecture

Johnson, A. and R. Wiggins, “Modeling the enterprise data architecture,” IBM Corporation web site

<http://en.wikipedia.org/wiki/Ontology>

Gruber, T., gruber@ksl.stanford.edu, “What is an ontology?”

Dedvedzic, V. (April 2002), “Understanding Ontological Engineering,” *Communications of the ACM* 45:4, pp. 136-144.

Lewis, G., et al. (2001, *An Enterprise Information System Data Architecture Guide*, CMU Technical Report ESC-TR-2001-018.

Katzan, H. (December 200), “Advanced Financial Management: Is there a framework in our future?” *INC Magazine*.

Sowa, John F. (2000), *Knowledge Representation: Logical, Philosophical and Computational Foundations*, Brooks Cole Publishing, Pacific Grove Publishing, Inc.

Stephens, T. (2007), “Why is Data Architecture so darn hard to implement?” www.BellSouth.com.